Partitioning and Relationship Detection for Large-Scale Semantic Graphs

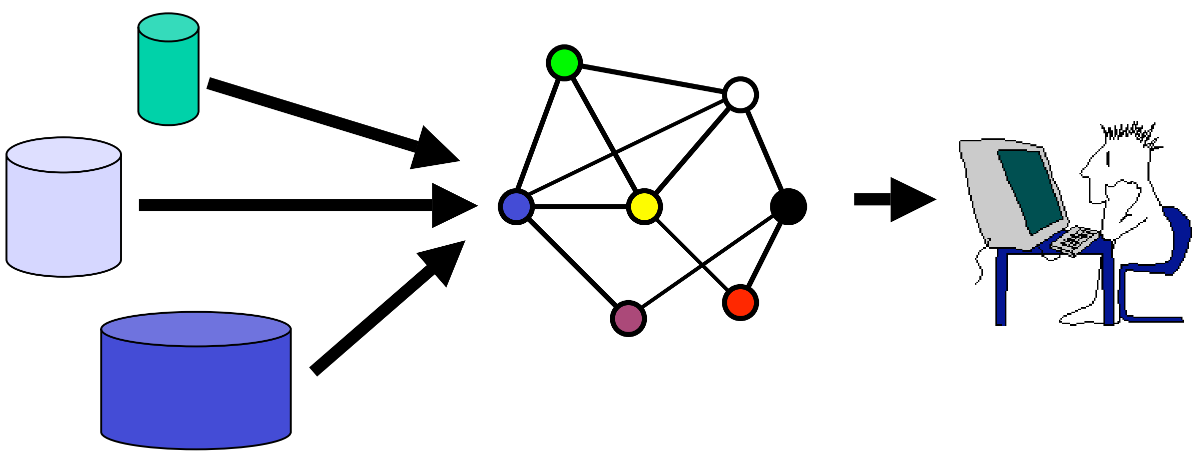
DHS Advanced Scientific Computing Program

Edmond Chow

Lawrence Livermore National Laboratory

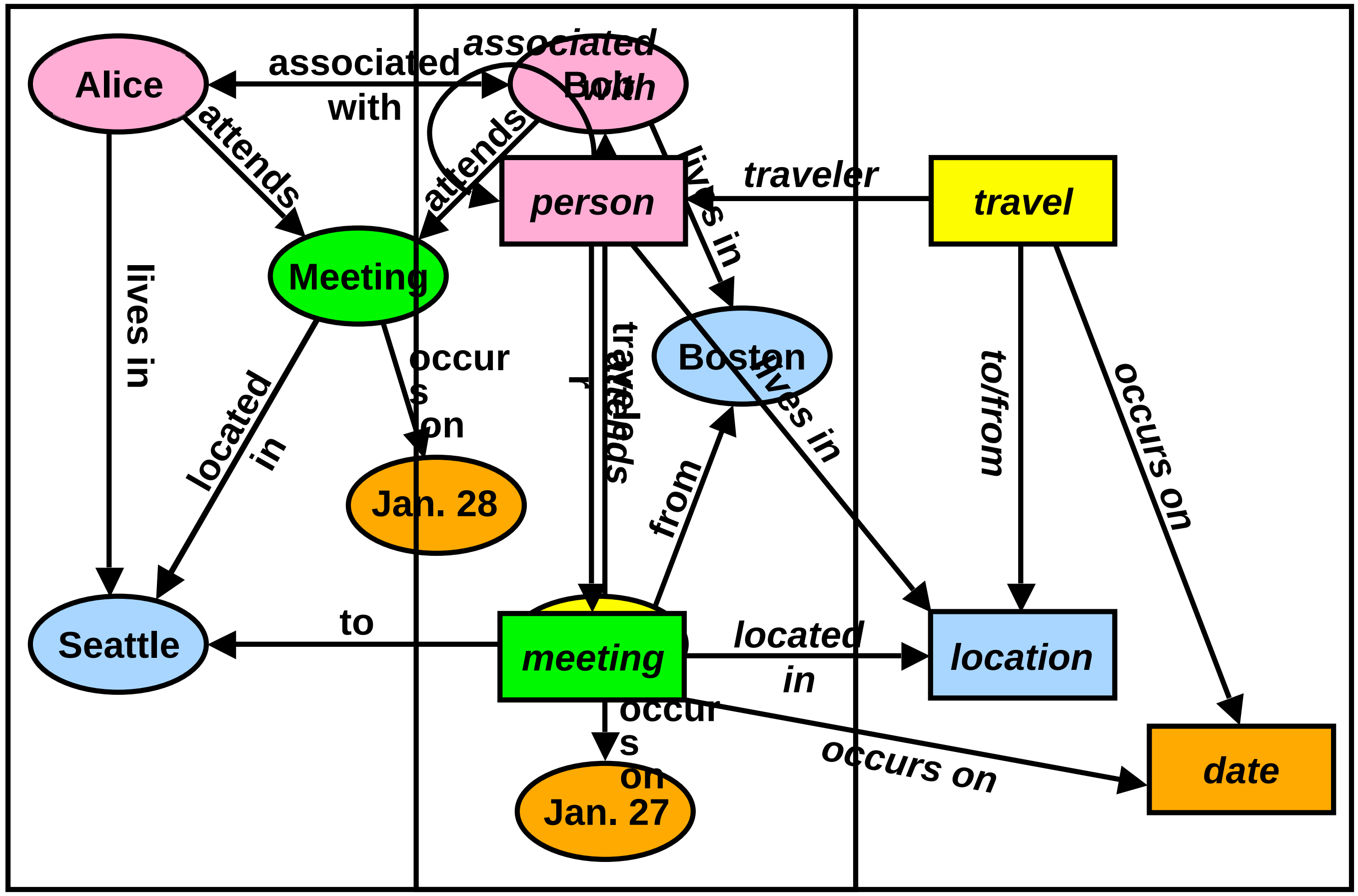
Motivation

- Intelligence analysts must identify relationships in huge amounts of data
- Data is collected from multiple sources at increasing rates
- Challenge: identify relationships and uncover patterns in a timely manner
- Approach: use *semantic graphs* to represent the data and graph algorithms to discover hidden relationships



Semantic graphs have attributes and types on the vertices and edges

Semantic Graph Ontology

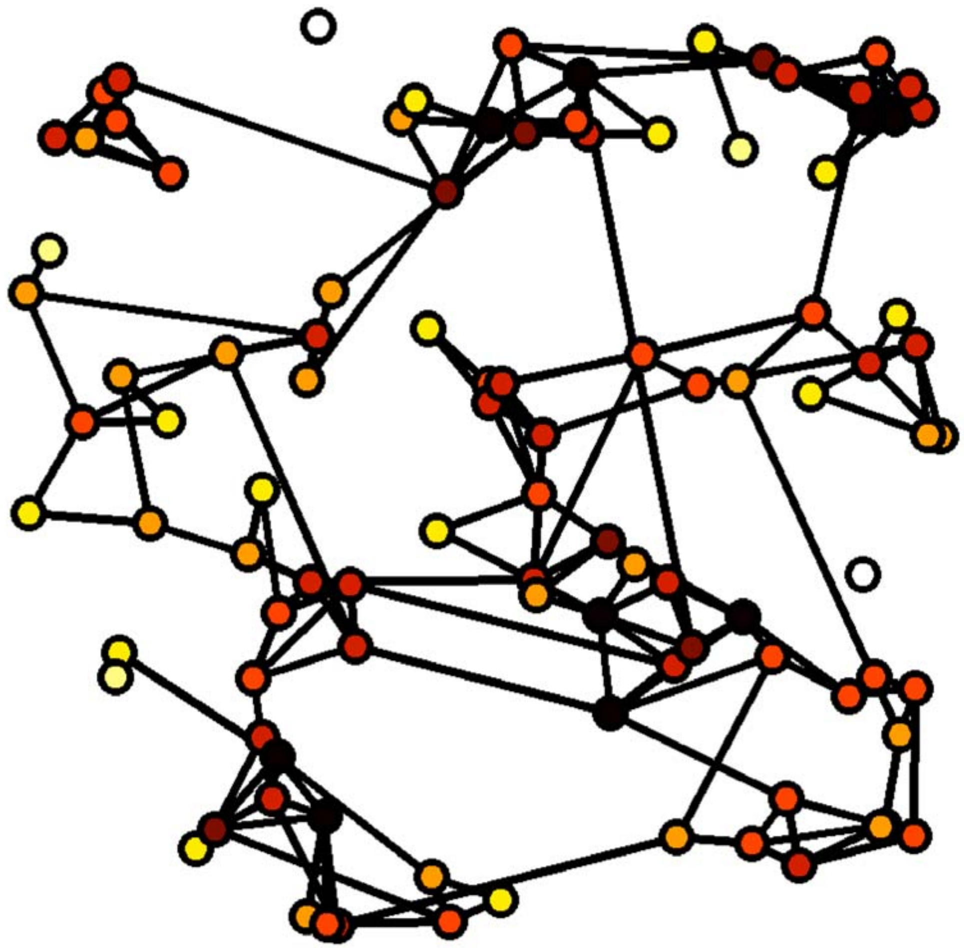


Semantic graphs for information analysis are becoming enormous

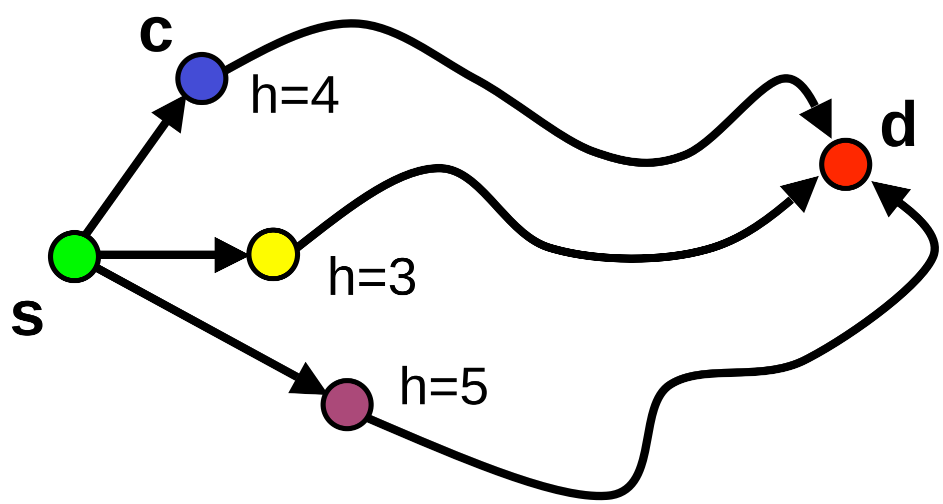
- Distributed memory parallel computers must be used to store and search these graphs
- Graphs must be partitioned onto separate memories and graph searches must have low communication cost
- Topological properties of semantic graphs make standard partitioning techniques ineffective
- Our goals are to develop partitioners and efficient, scalable parallel search algorithms

We are developing parallel algorithms to search massive graphs

- Partitioning for semantic graphs
- Heuristics for searching semantic graphs, incl. template matching
- Scalable parallel implementations
- Properties of complex information networks
- Knowledge discovery in relational data



To find the shortest path, use A* search, which uses a heuristic to guide the search

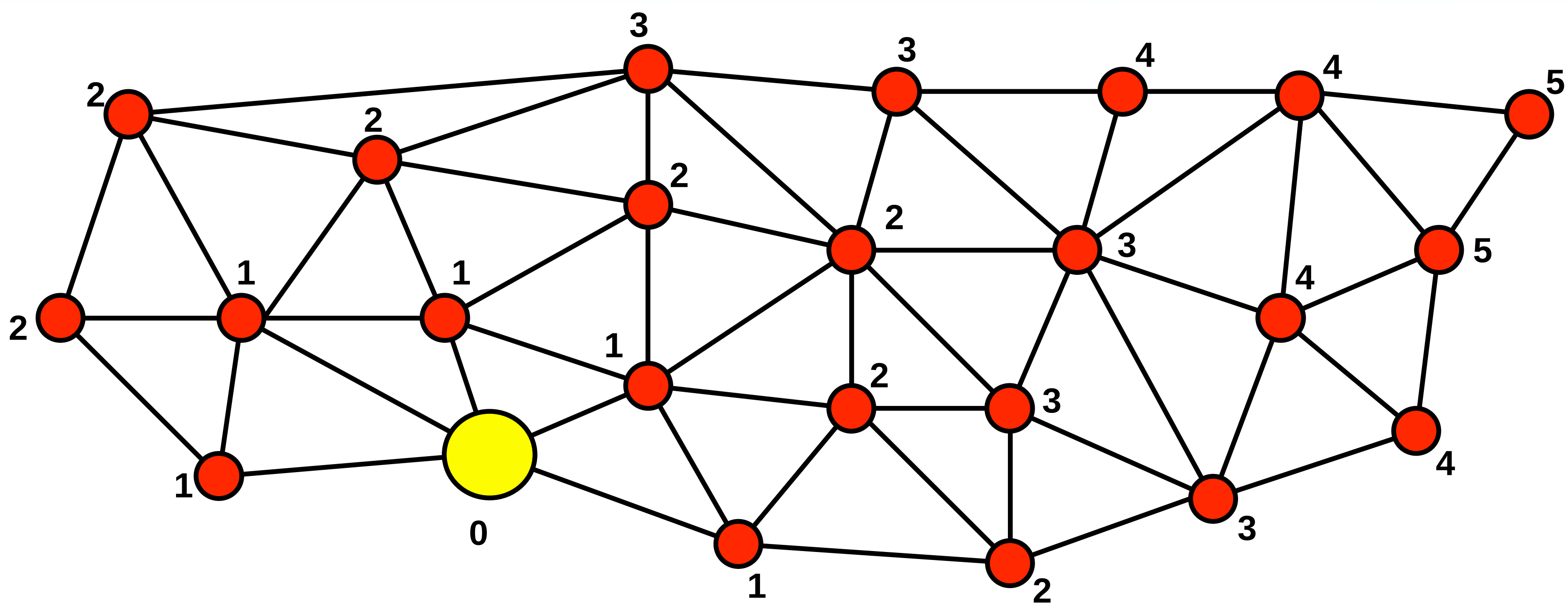


Use a cost function of the form:

f(s,c,d) = g(s,c) + h(c,d)

h is *admissible* if it never over-estimates actual cost

Level-difference heuristic

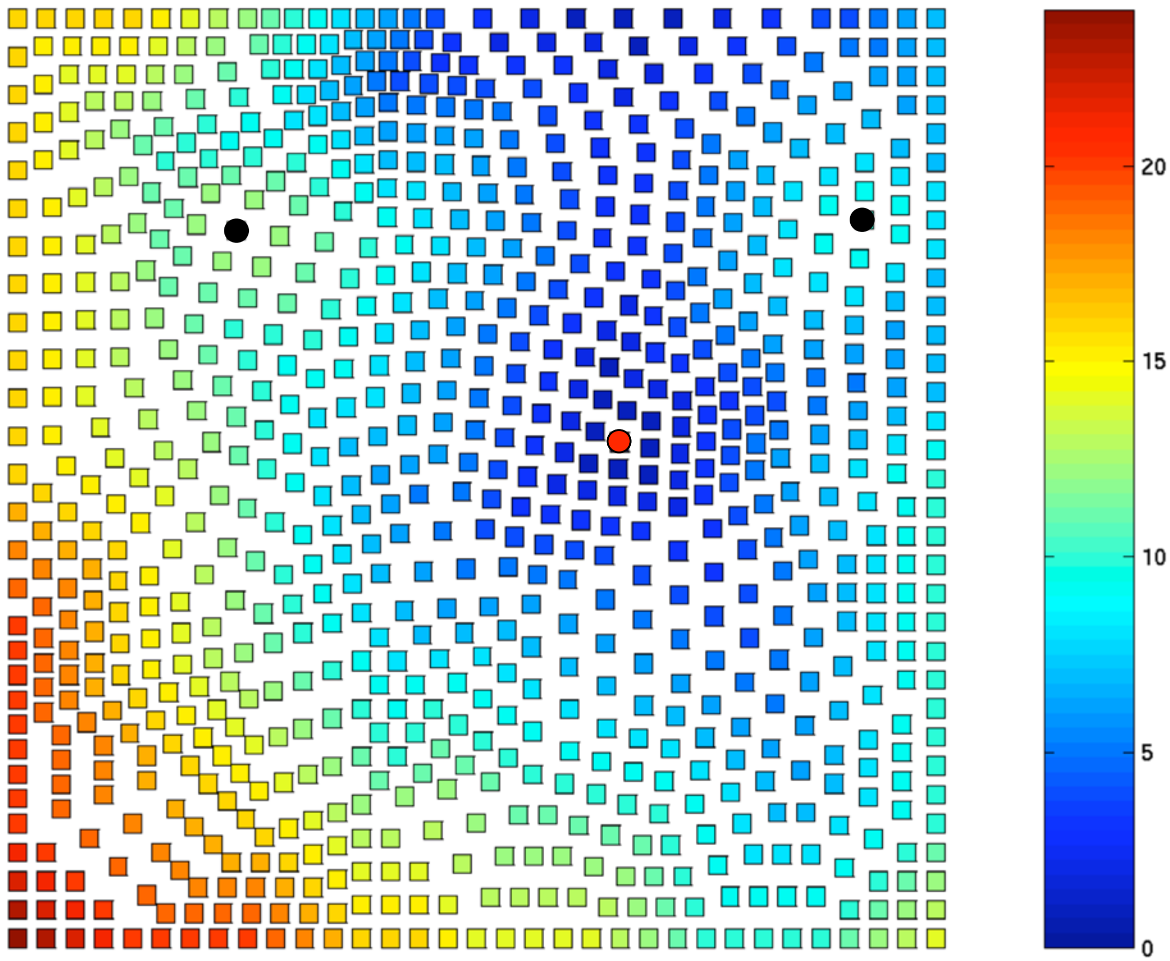


$h_1(i,j) = |\ level(i) - level(j)\ |$

$h(i,j) = \max\{\ h_1(i,j), \ldots, h_m(i,j)\ \}$, m centers

Error in the heuristic can be bounded

Example: heuristic distance to a vertex on a mesh

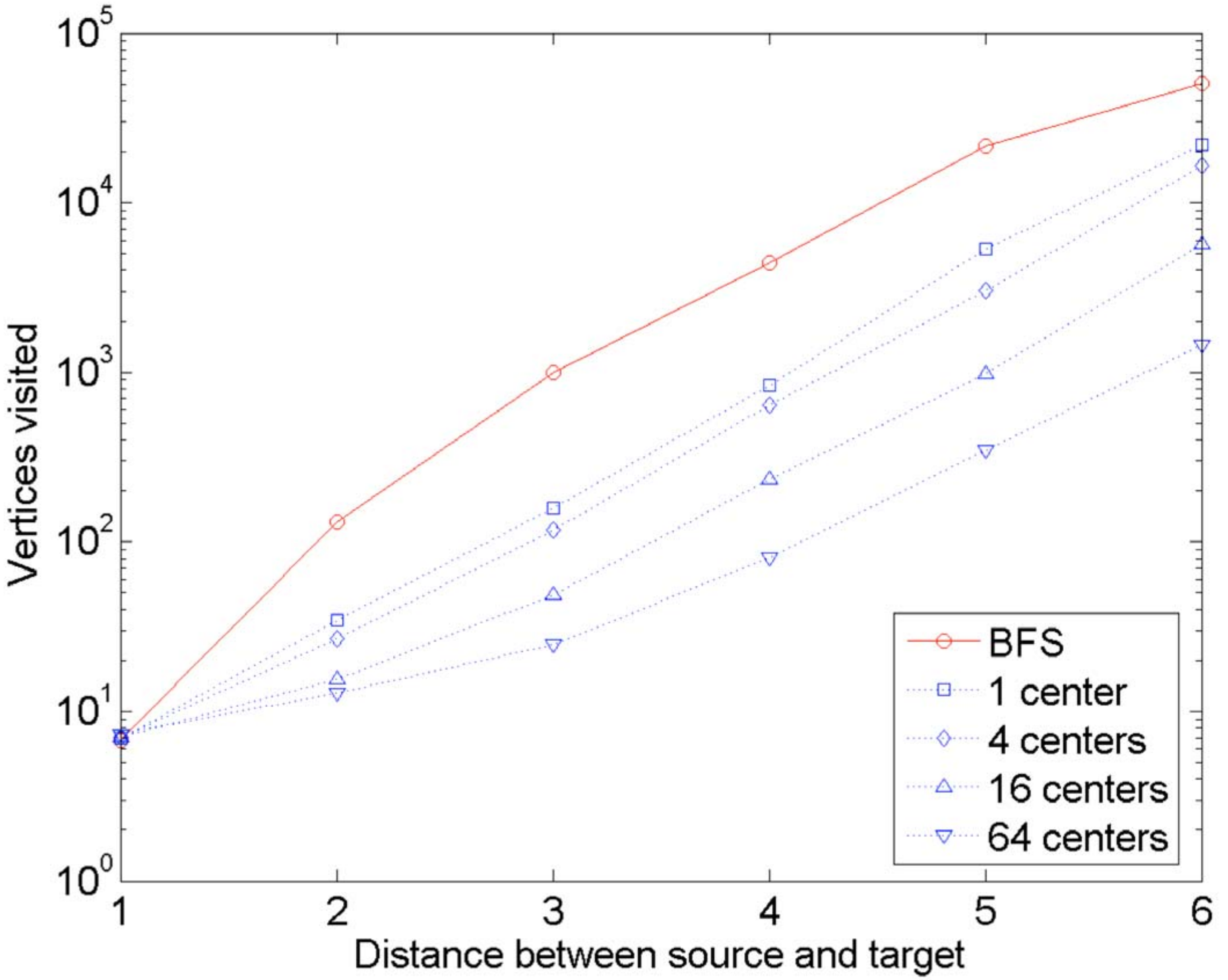


Heuristic distance to red vertex given two center vertices

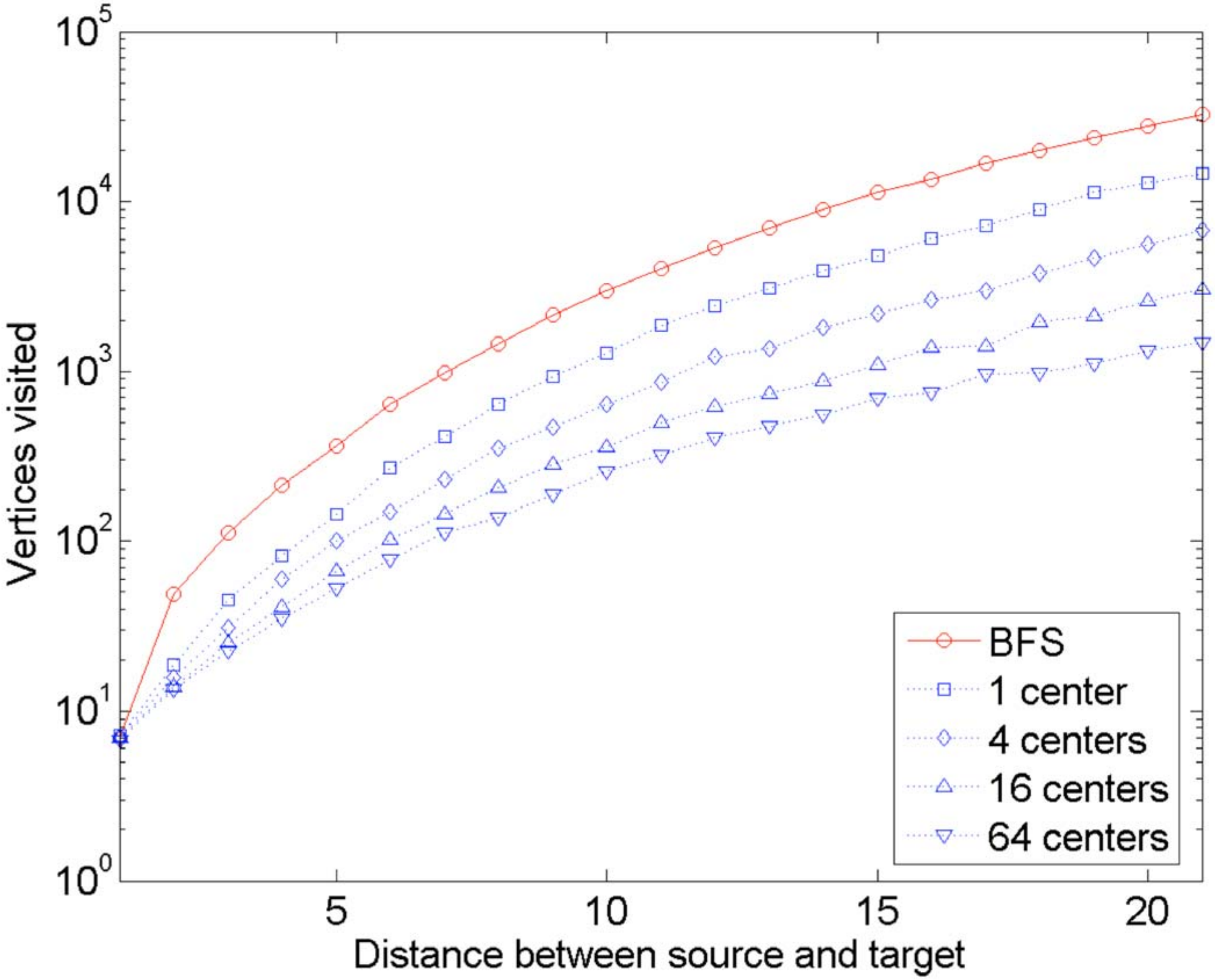
Test graphs

Type	Vertices	Edges	Ave Deg	Ave Path Length	Diameter
Random	63848	191999	6.0	6.4	12
Random	51065	61415	2.4	15.0	39
Random	127664	383997	6.0	6.8	12
Spatial	64000	192000	6.0	21.3	39
Mesh	64304	192043	6.0	116.4	278
Internet	112969	181639	3.2	9.9	27

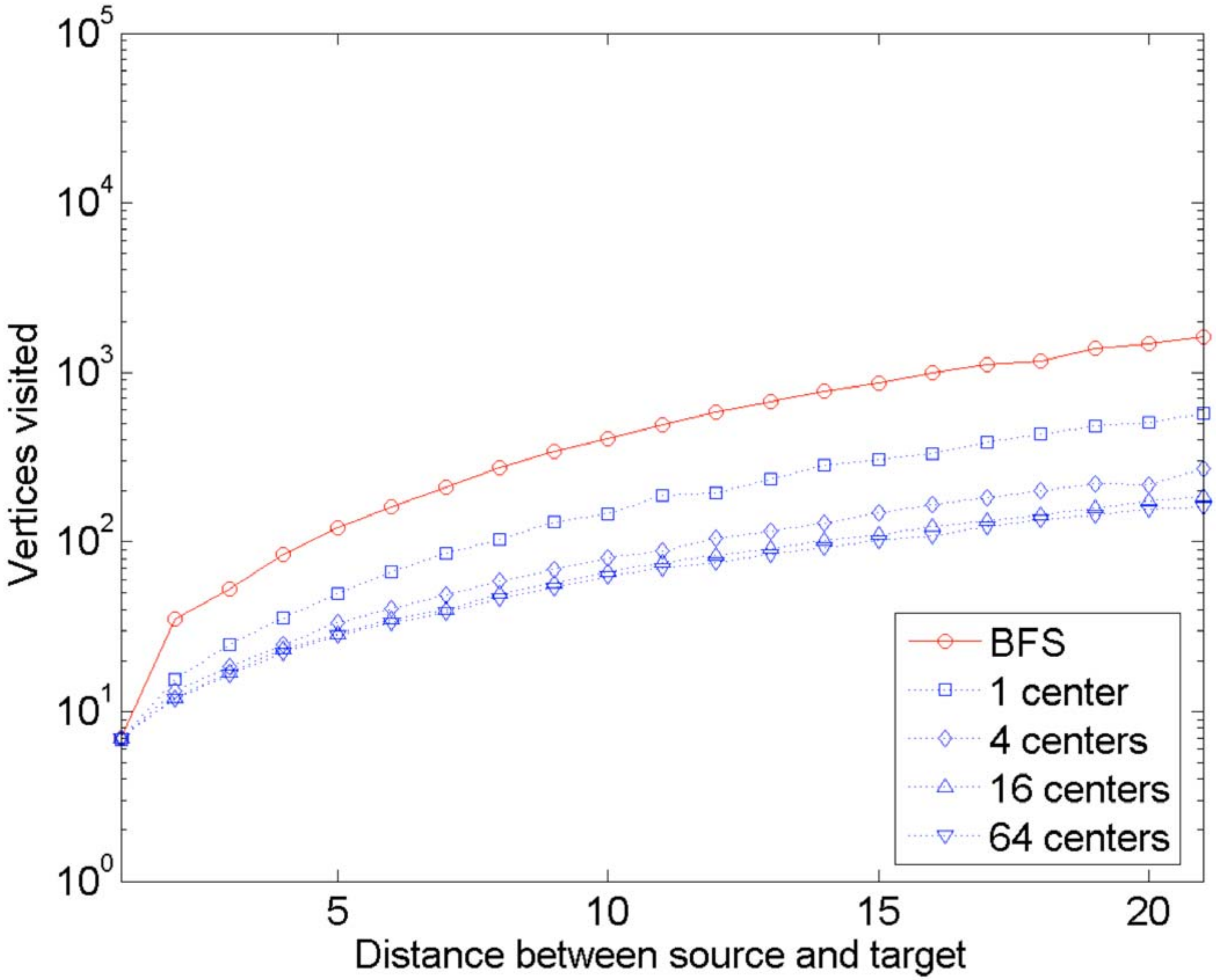
Random graph, 64k vertices, <k> = 6



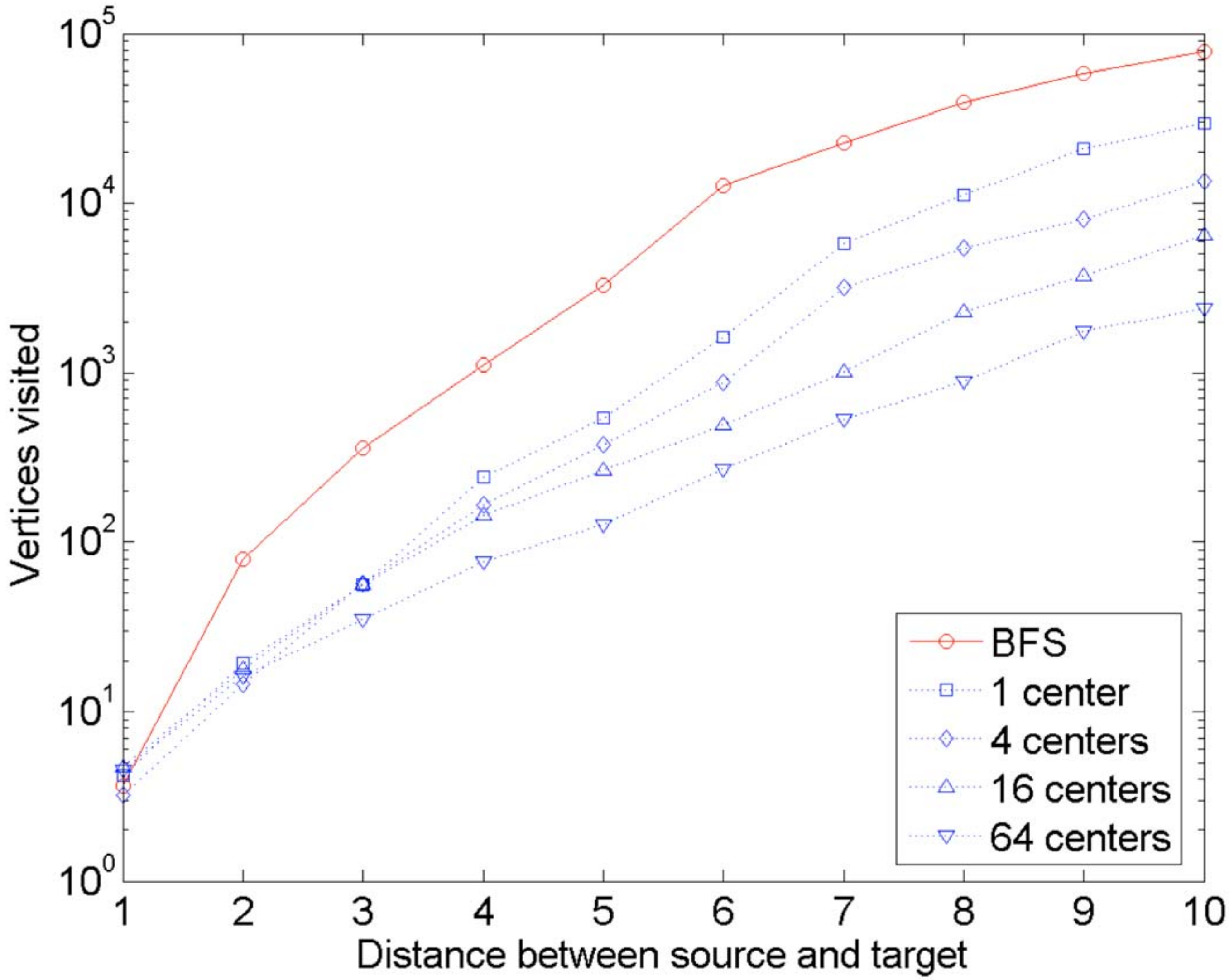
Spatial graph, 64k vertices, α=4



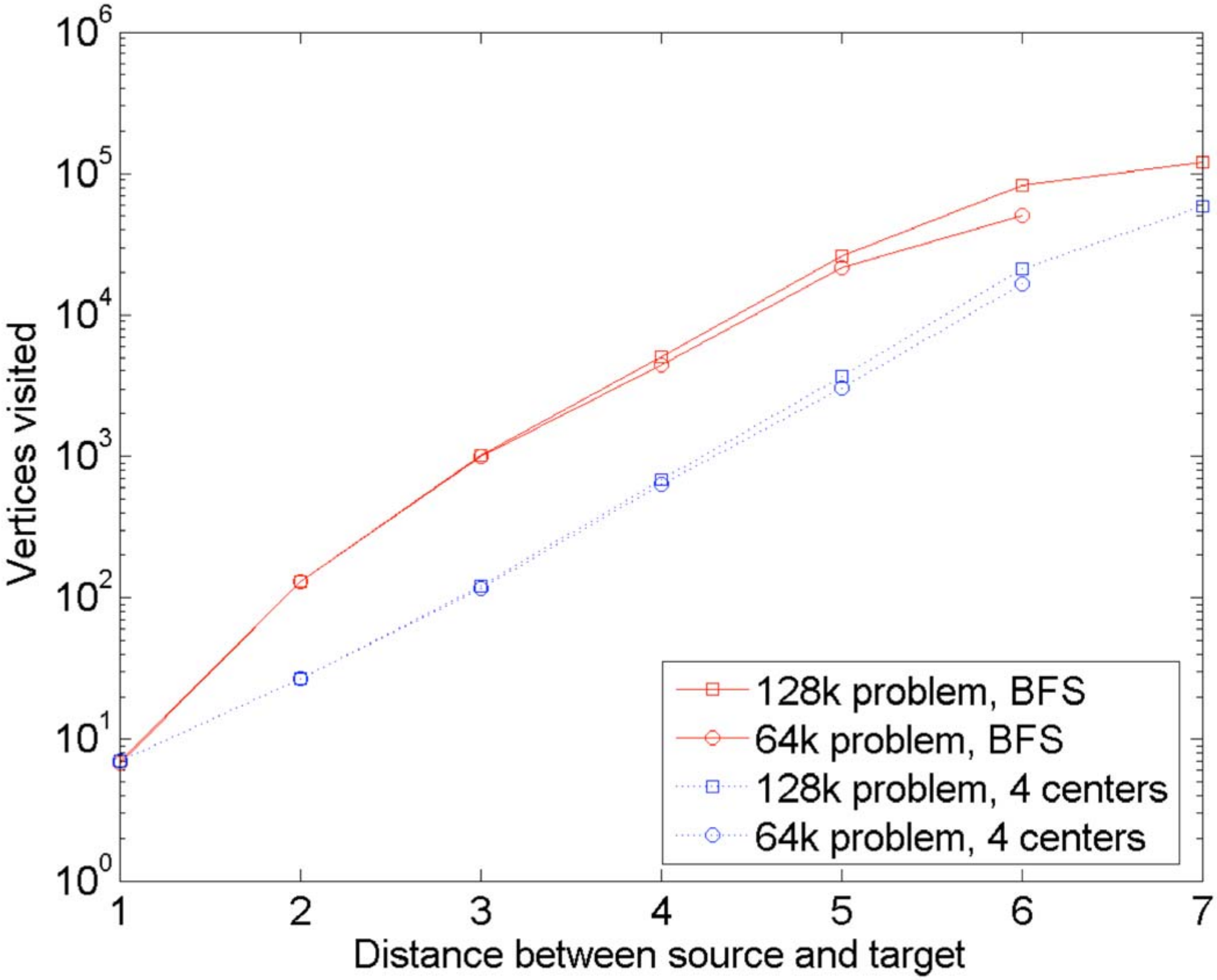
2-D Mesh, 64k vertices



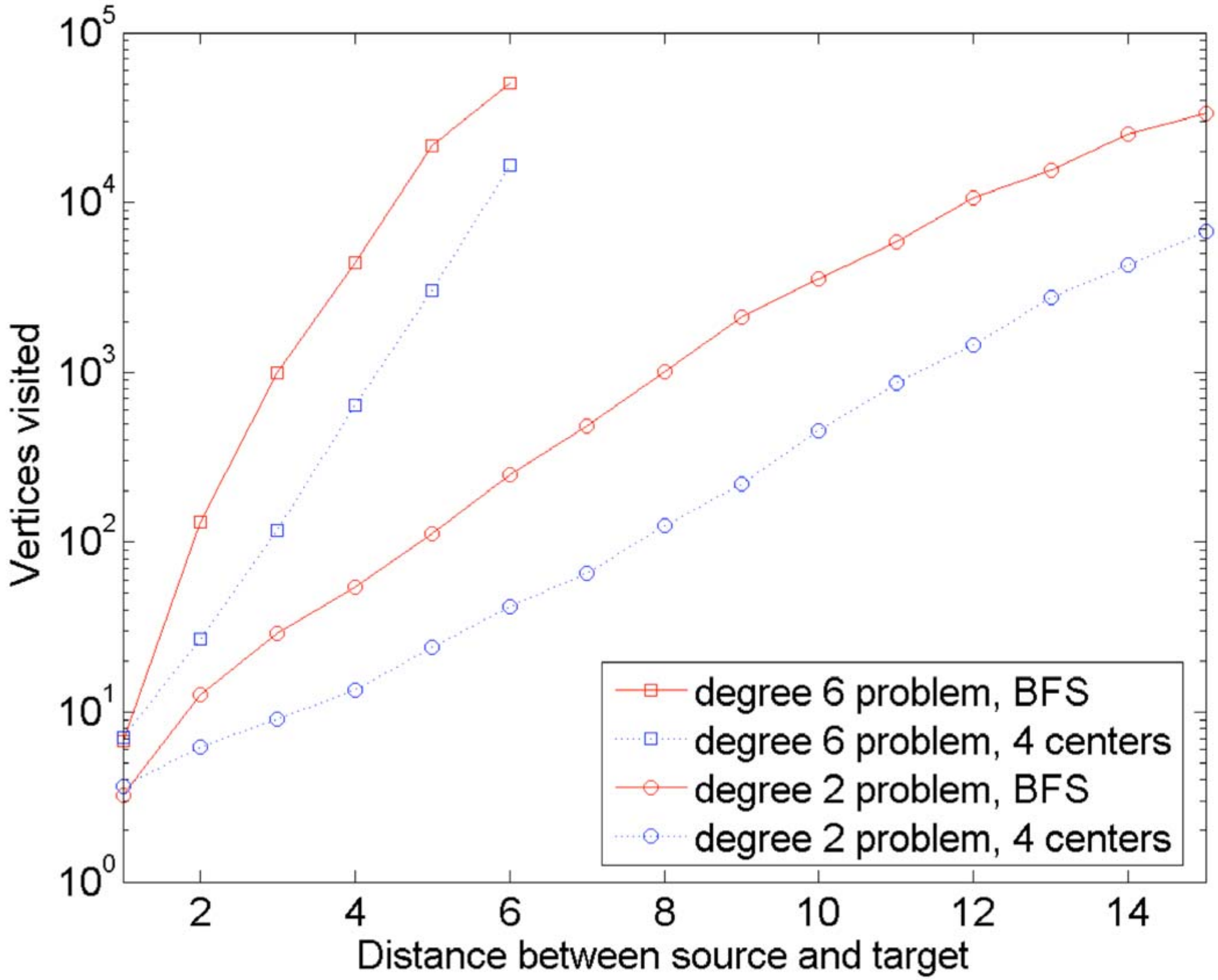
Internet graph



Scalability (random graphs <k> = 6)



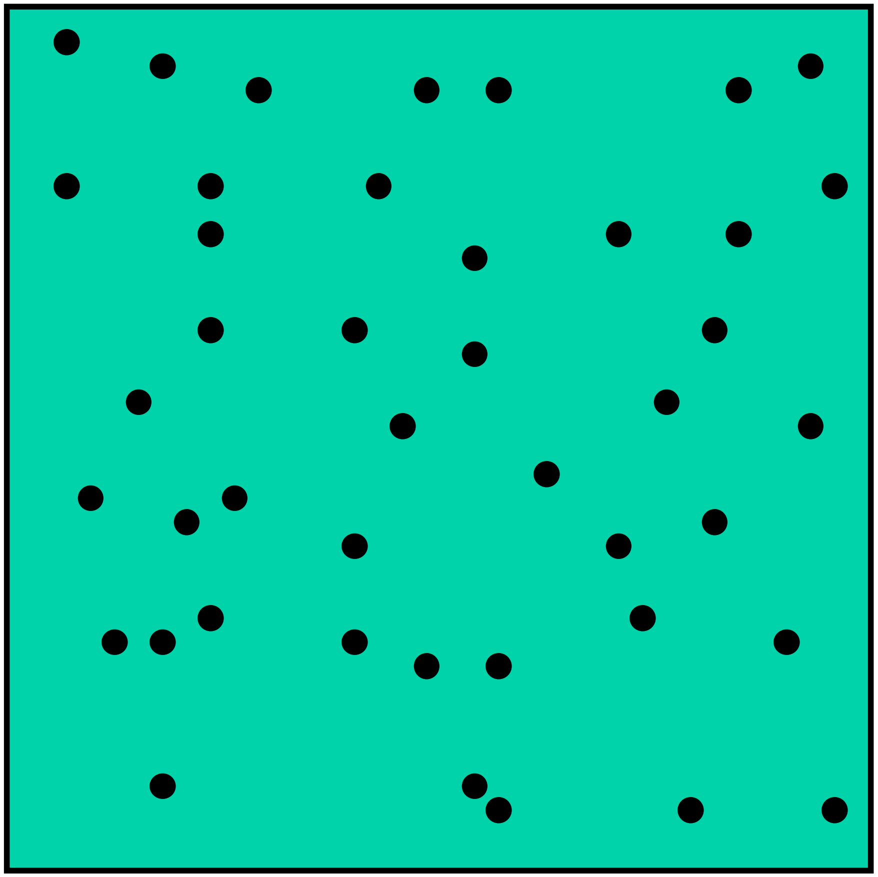
Effect of vertex degree (random graphs)



2D Partitioning

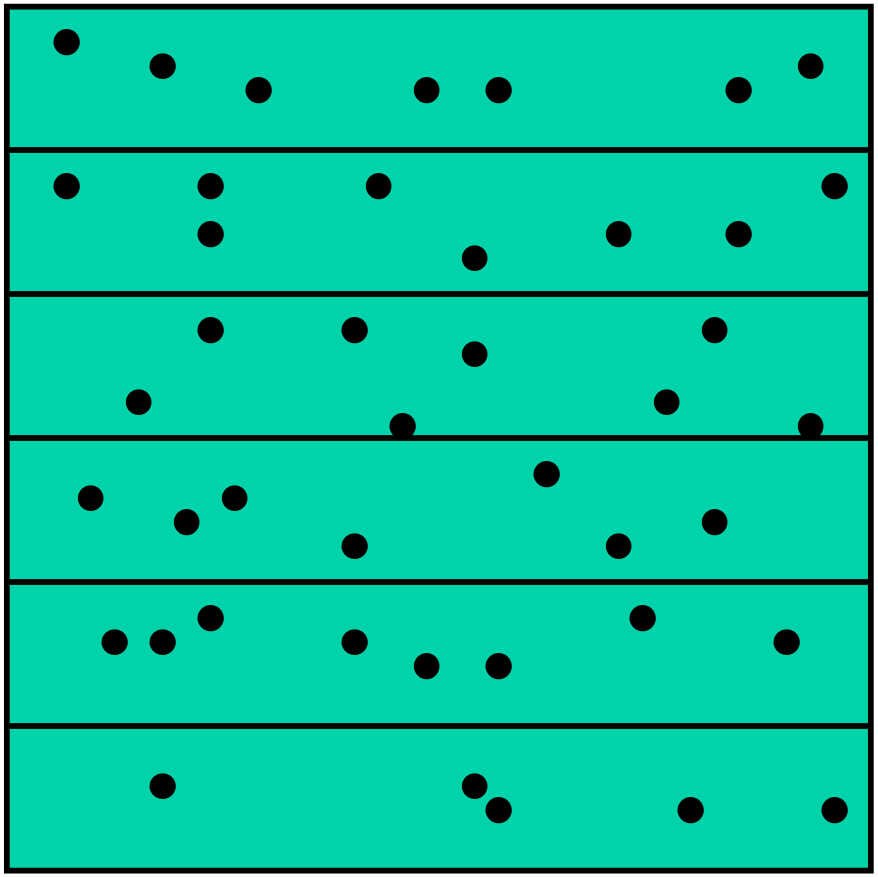
- Partition the vertices (1D) or the edges (2D)
- 2D partitioning has been advocated for sparse matrices where the sparsity pattern is difficult to exploit (Hendrickson, Leland, and Plimpton 1995)
- Many variants of 2D partitioning (Catalyurek 1999)
- 2D checkerboard variant is perhaps most useful
  - Redistribution-free, transpose-free doubling/halving (Lewis and van de Geijn 1993, Lewis, Payne, and van de Geijn 1994)
  - 2D checkerboard (Catalyurek 1999, Catalyurek and Aykanat 2001)

Example: Adjacency matrix

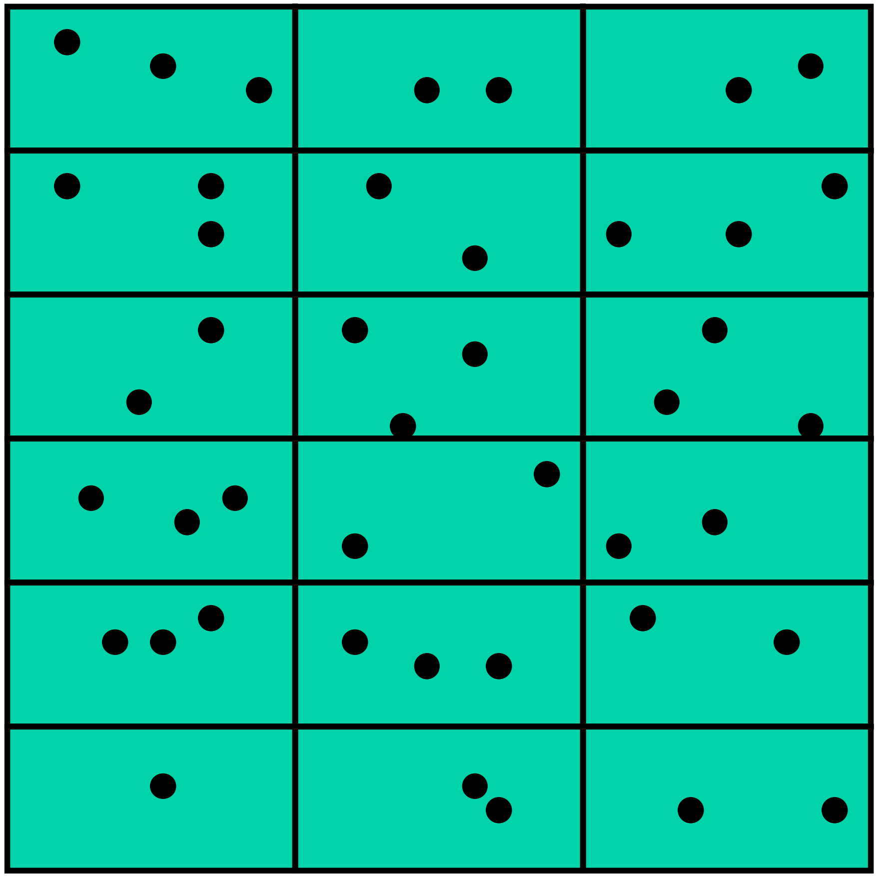


Partition to minimize processor communication while maintaining load balance

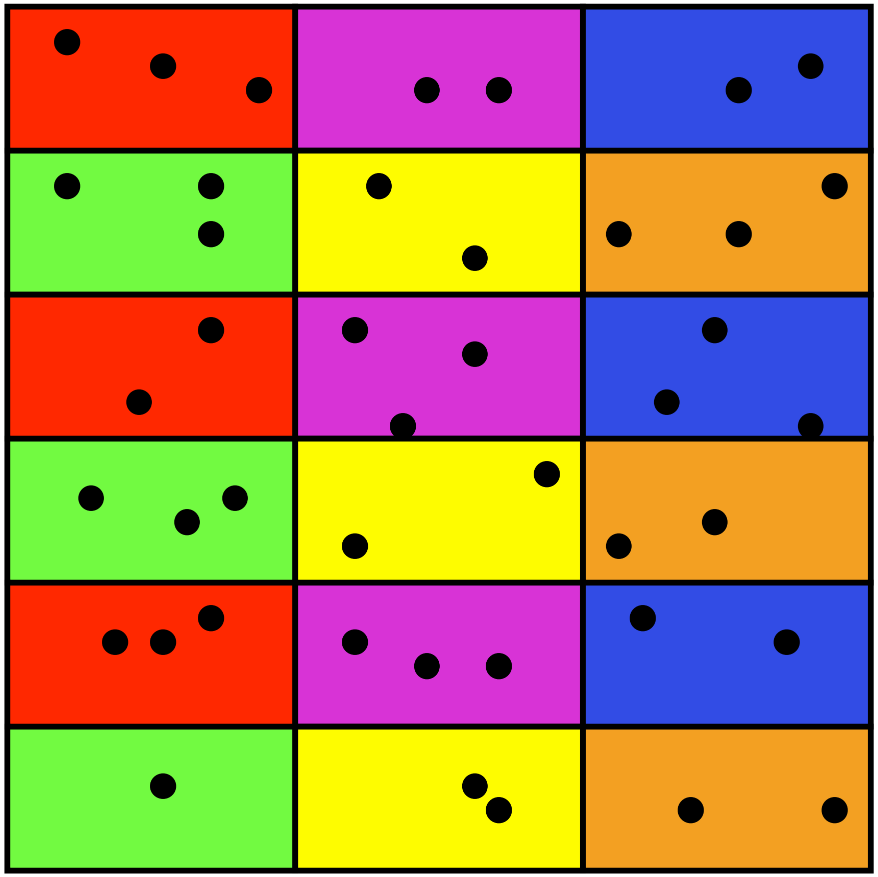
Example: 6-way Vertex Partitioning

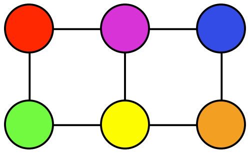


Example: 2x3 Edge Partitioning



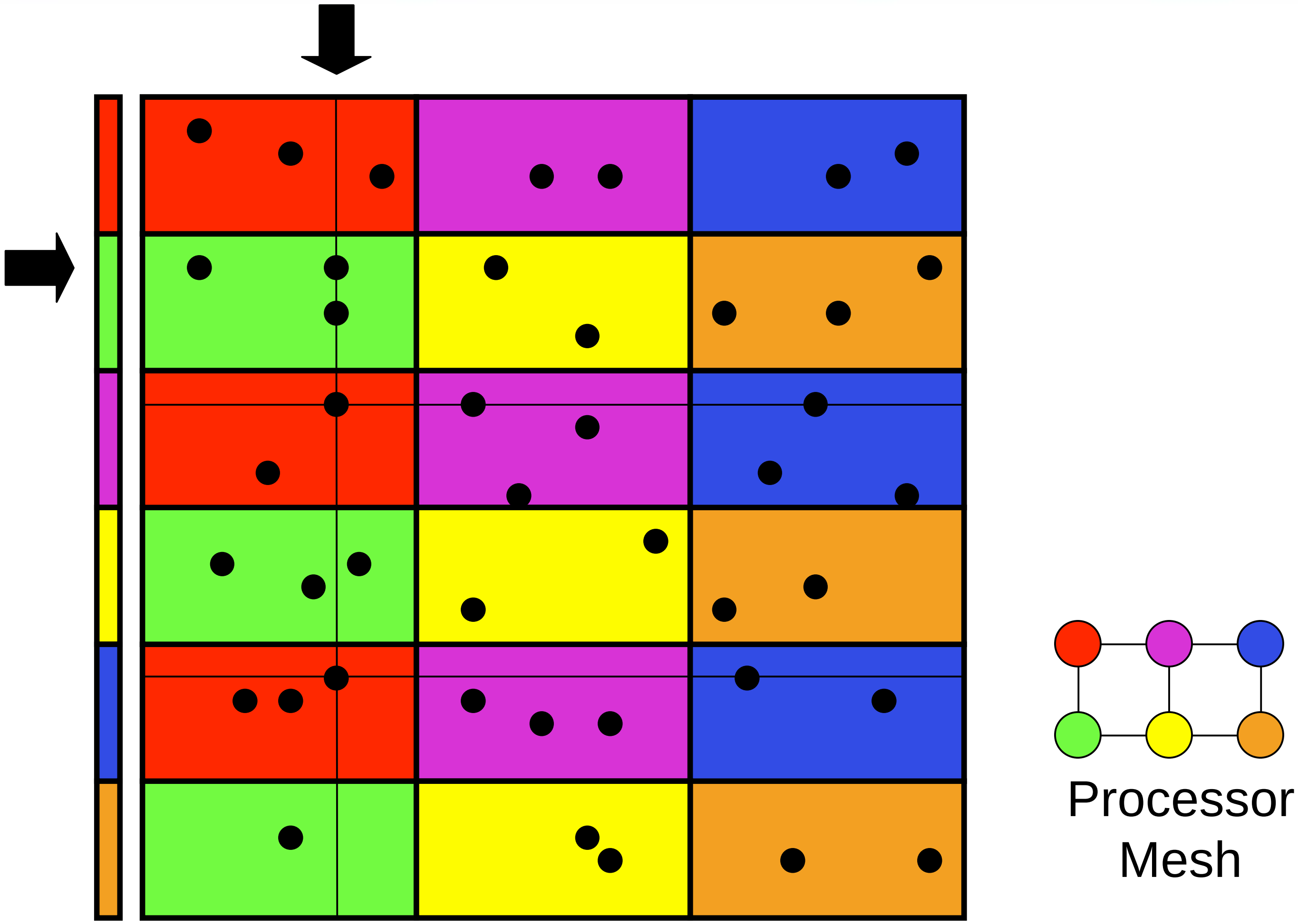
Example: 2x3 Edge Partitioning





Processor Mesh

Example: 2x3 Edge Partitioning



Level-synchronized Parallel Search

Do l=0 to … until target is found

F = set of assigned vertices with level l

Column Expand communication (send F, receive F')

N = set of neighbor vertices of F'

Row Fold communication (send N, receive N')

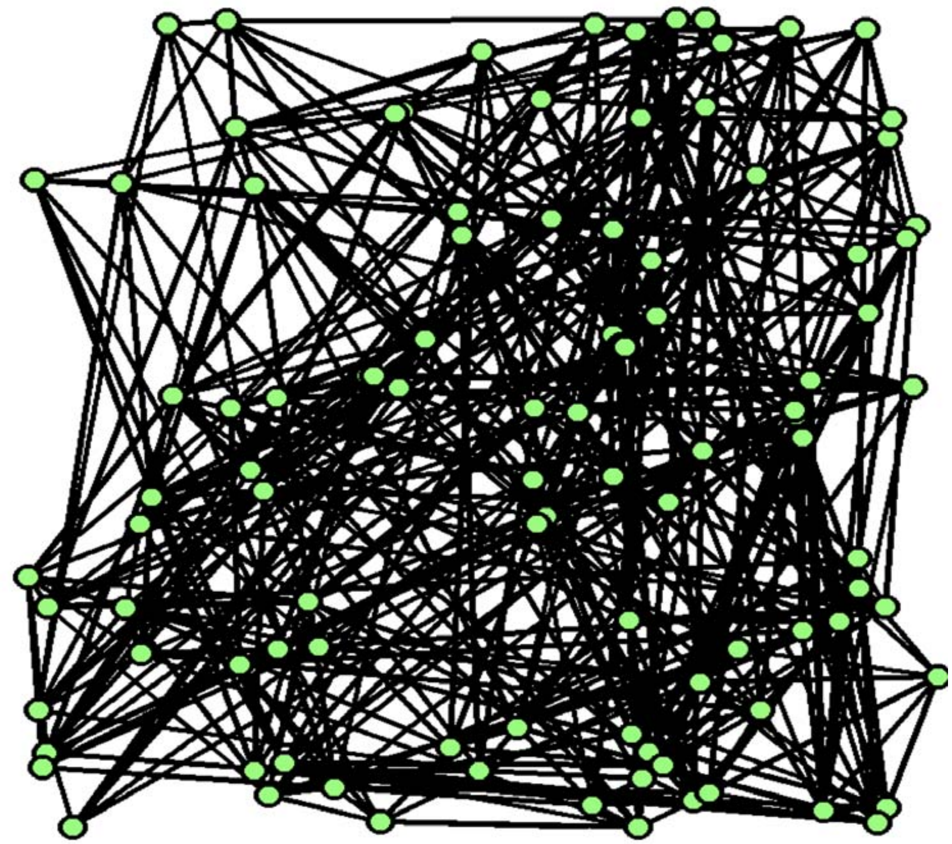
Update levels of vertices in N'

Enddo

- Expand is all-gather or all-to-all
- Reduce is all-to-all or reduce-scatter
- Must store *vertex* lists in sparse mode
- Storage is scalable for random graphs
- If the blocks are balanced, then the communication is balanced for any graph

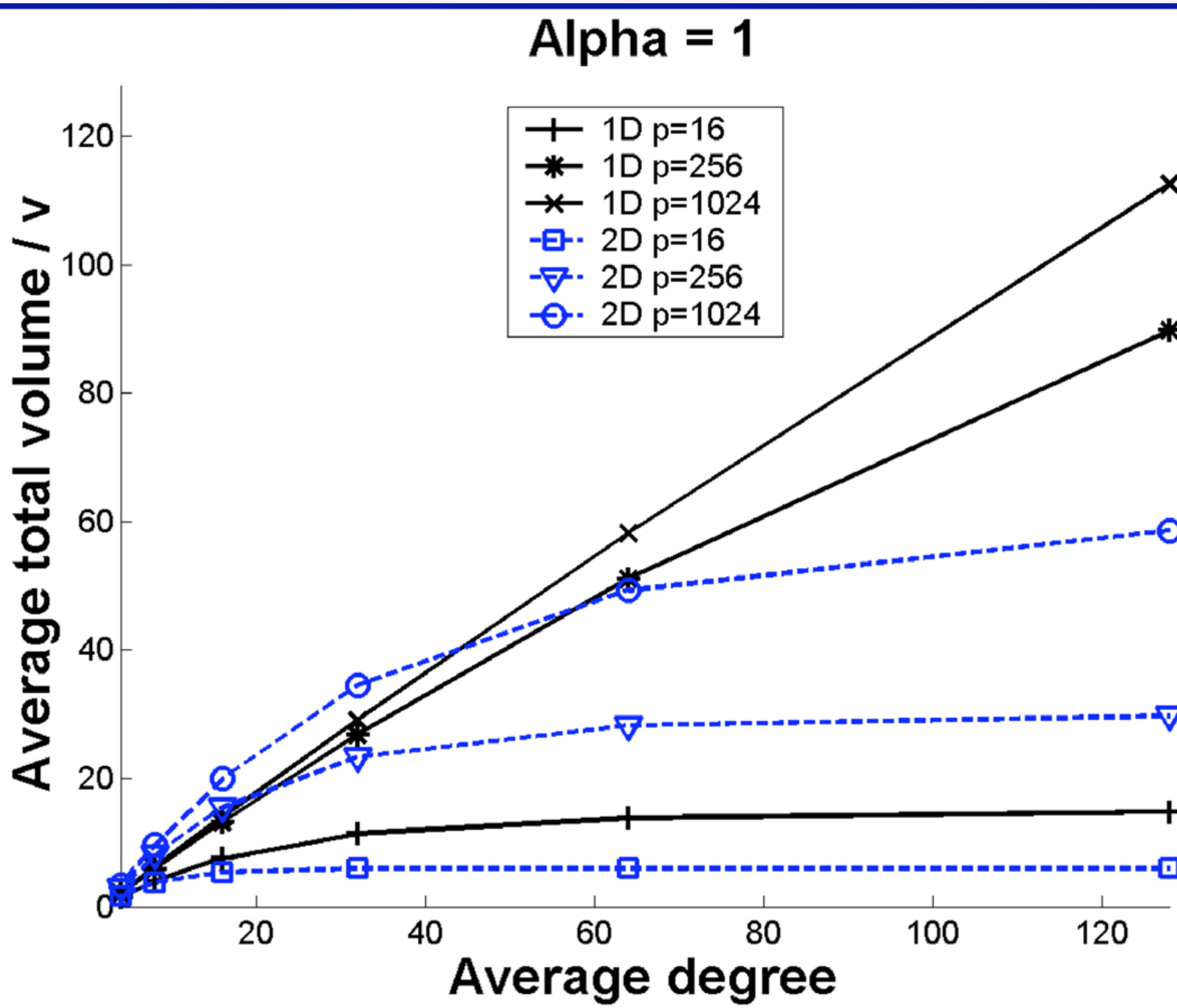
Spatial networks

- P(edge) ~ length(edge)$^{-\alpha}$
- Poisson random graphs have α = 0
- α is related to clustering coefficient
- Best partitioning is geometric



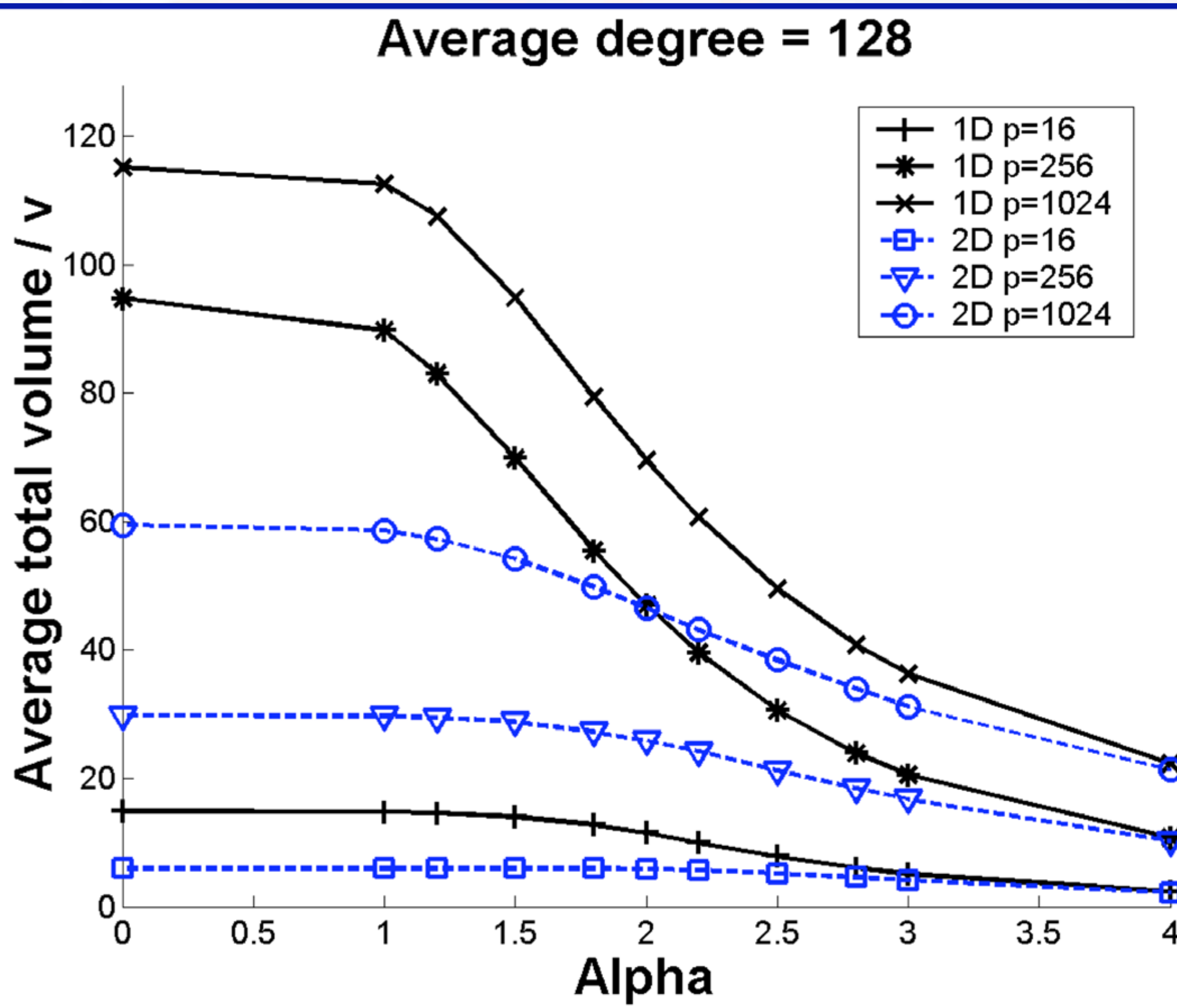
Spatial network with α =1 and avg. degree 10

Communication volume for 1D and 2D partitioning

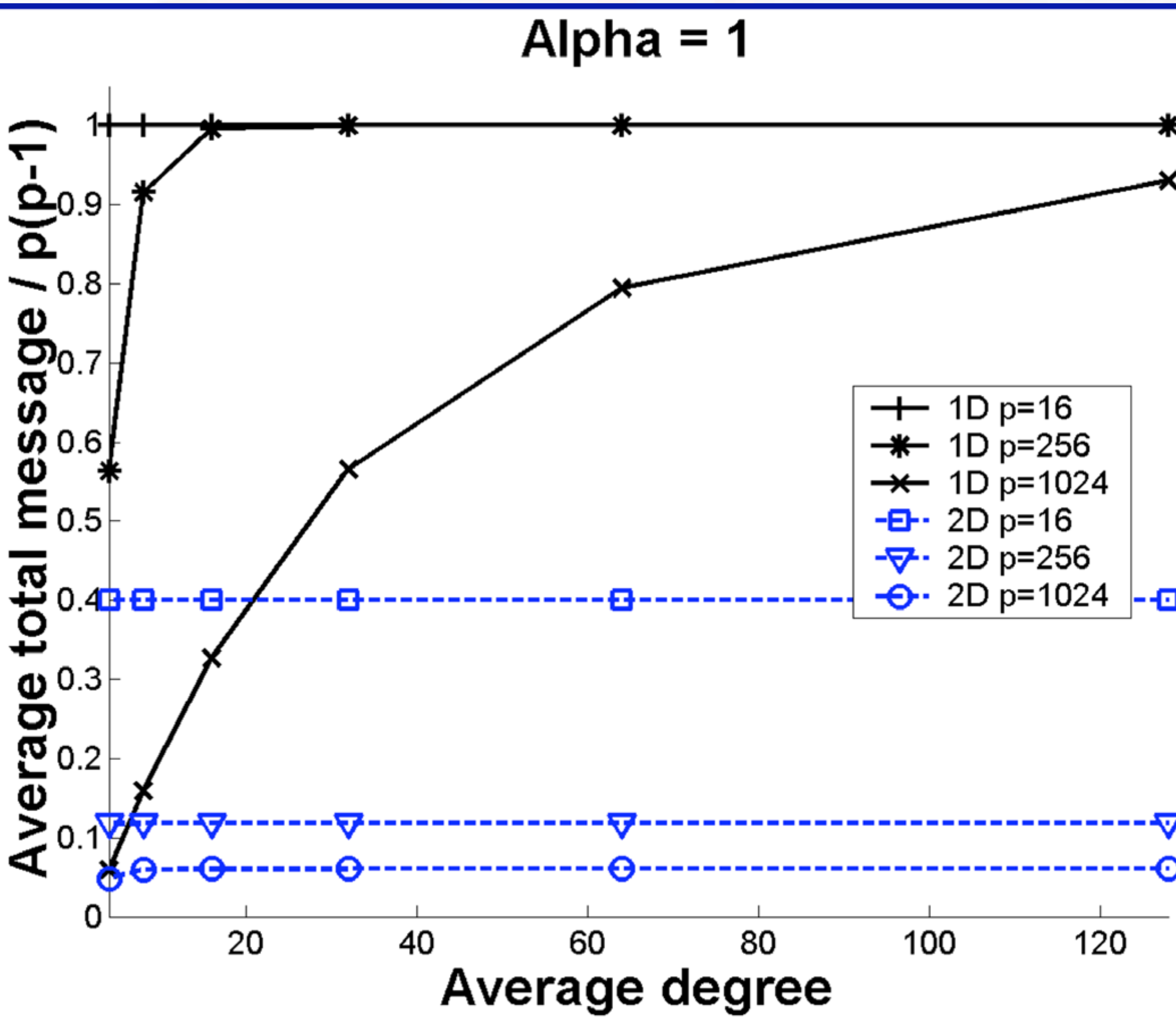


Compute
d
by PaToH

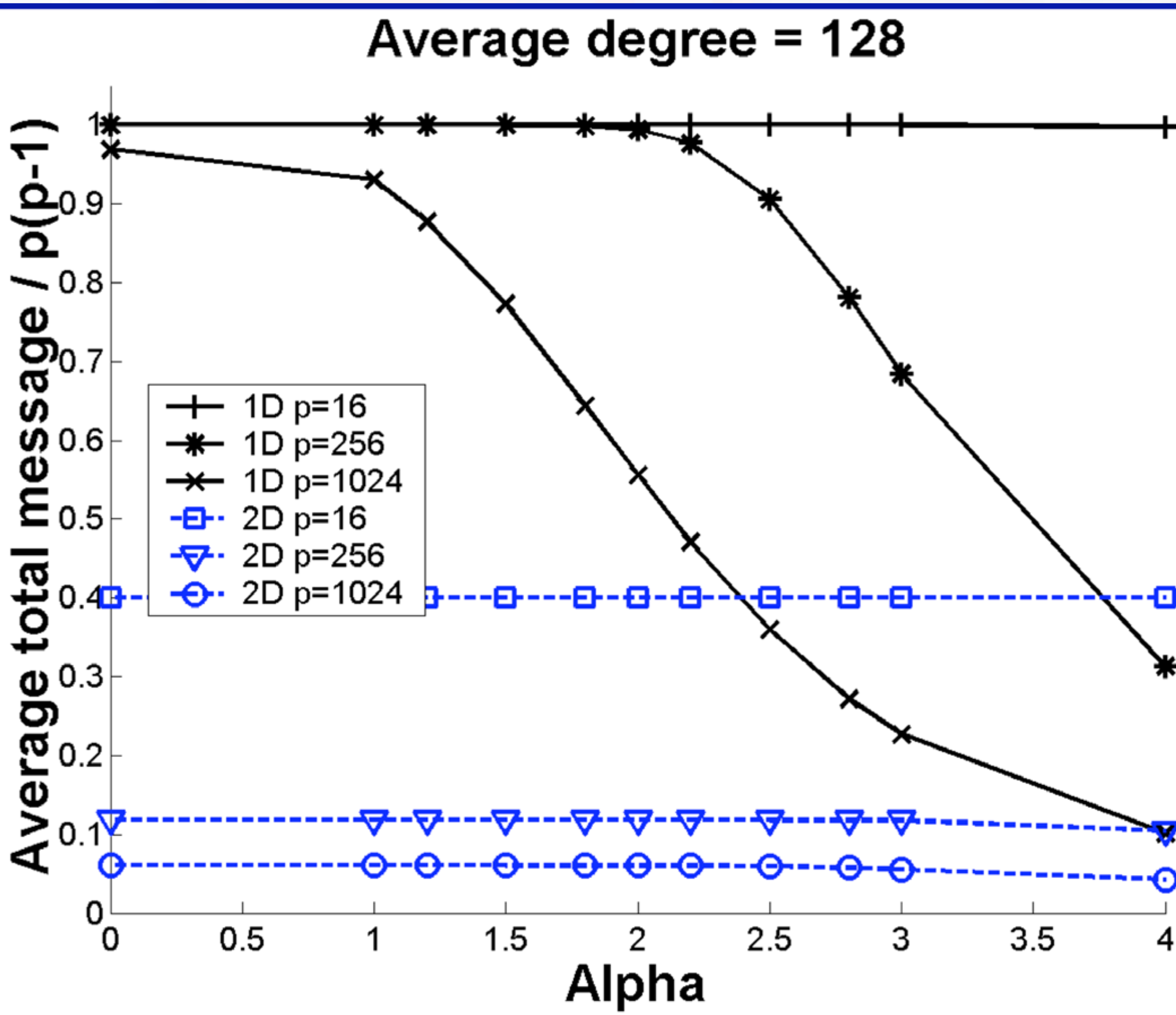
Communication volume for 1D and 2D partitioning



Number of messages for 1D and 2D partitioning



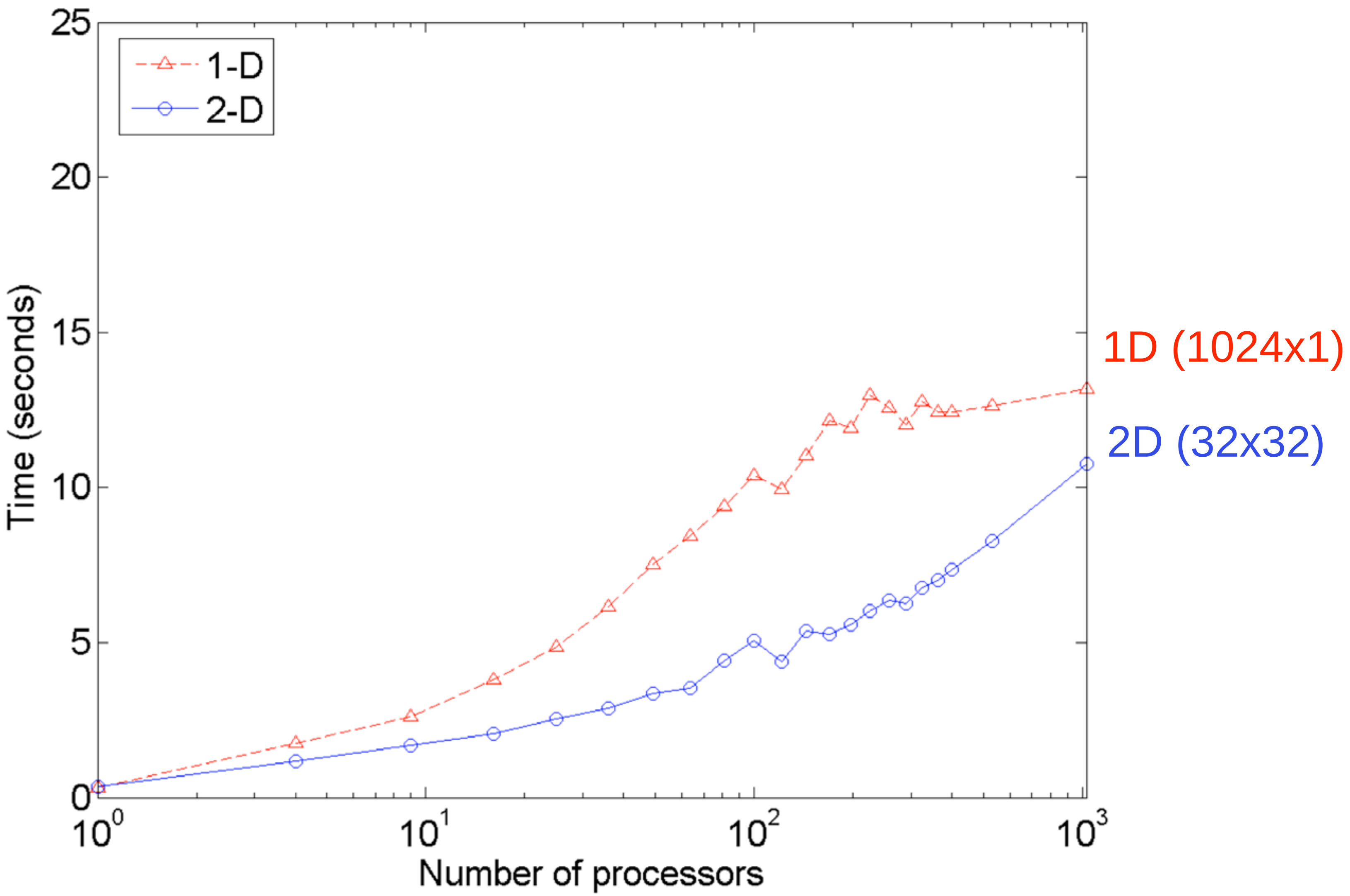
Number of messages for 1D and 2D partitioning



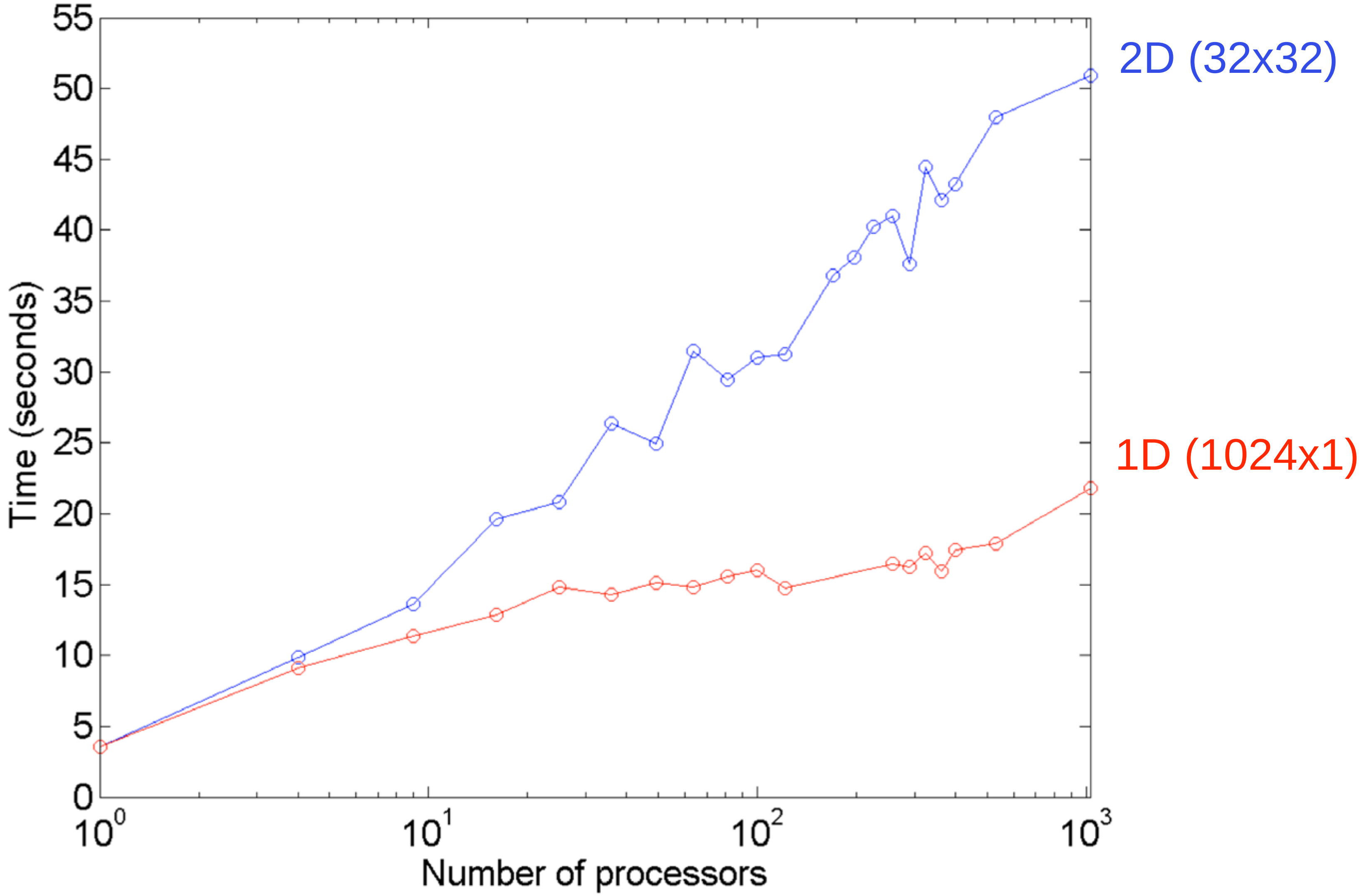
Parallel Search Experimental Setup

- Parallel Breadth First Search (BFS) algorithm
  - Level-synchronized algorithm
  - Report average time for 100 pairs
  - Does not take into account increasing graph avg. path length (varies from 5 to 9)
- Input graphs
  - Undirected Poisson random graphs with degree 10 or 100
  - Random 2D checkerboard partitioning
  - Vertices and edges accessed from memory
- Machines
  - MCR (Quadrics Linux Cluster)
  - BlueGene/L

Weak-Scaling, up to 1024 processors, <k>=100, 100 million vertices, 10 billion edges



Weak-Scaling, up to 1024 processors, <k>=10, 1 billion vertices, 10 billion edges

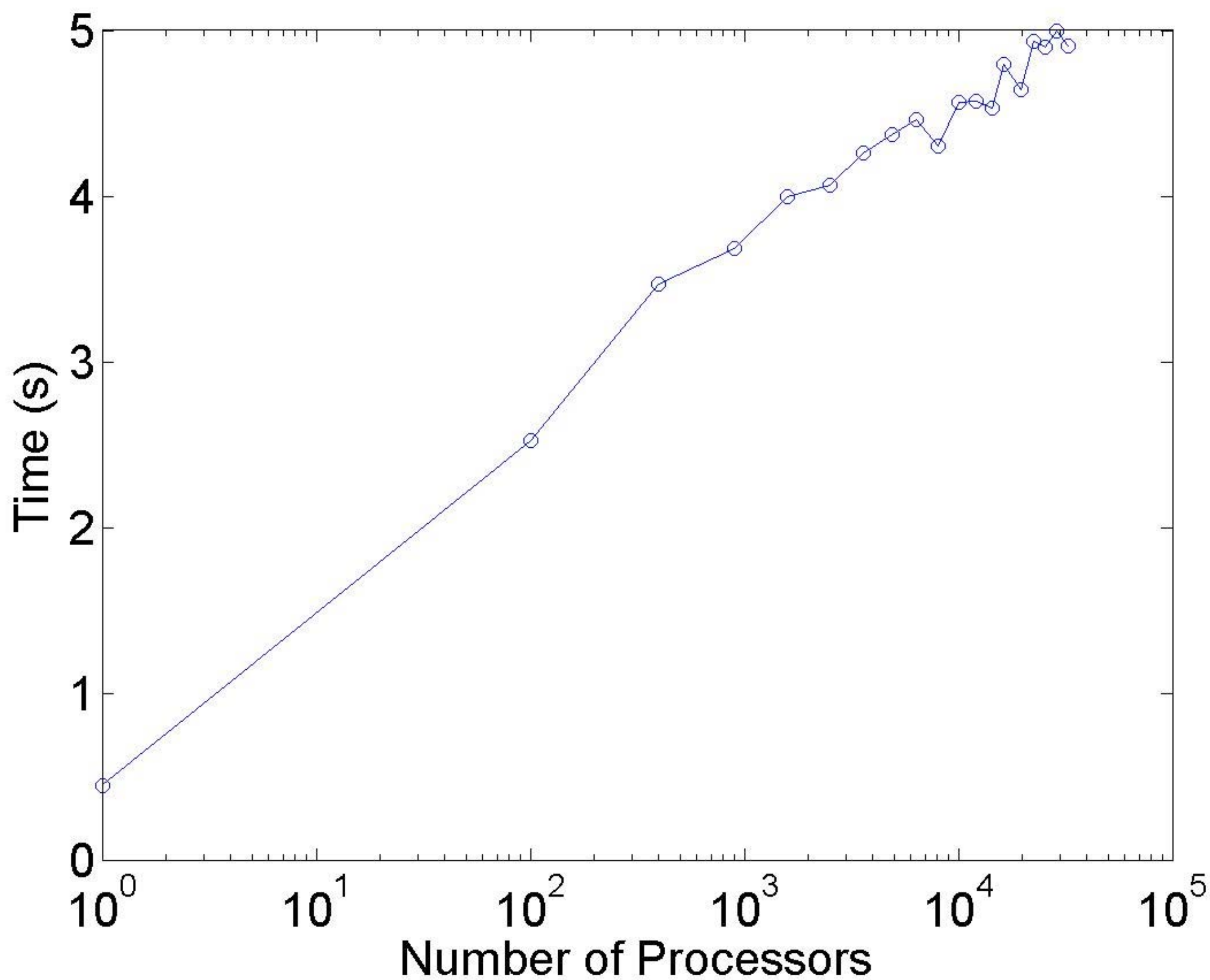


BlueGene/L timings, up to 32k processors

Number of Vertices	Processor Mesh	Search Time (s)
1.00 Billion	100 x 100	4.37
1.96 Billion	140 x 140	4.64
3.28 Billion	181 x 181	4.90

Constant local problem size of 100k vertices/processor for a random graph with average degree 10.

Scalability on BlueGene/L up to 32k processors



Conclusions

- Heuristic search can be used to reduce the cost of relationship detection
- 2-D partitioning is effective for unstructured graphs with high average degree
- For more information: http://www.llnl.gov/casc/compnets

Project Team and Collaborators

- Edmond Chow
- Tina Eliassi-Rad
- Keith Henderson
- Andy Yoo

- Bruce Hendrickson and William McLendon (Sandia Natl Labs)
- Umit Catalyurek and Doruk Bozdag (Ohio State University)

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.